

APPENDIX A

Programming HOS (Hybrid Operating System)

The Hybrid Operating System is designed to provide for a low core program with the library as a top priority programmed interrupt. It supplies all inputs and outputs and error checking of the disks, printer-reader, and punch equipment. It is also designed to provide continuity between the three programs, since the 1401 is constructed to run only one program at a time. It also provides for on-line processing of the 1030 data terminal located in the library.

The MONITOR Control Program uses 150 core storage positions. The program starts with "MONIN" (a symbolic label for point of entry) and is instructed to save the address just branched from, for later use. It checks the I/O with the next sequential instruction from the User Program and stores it in the "MONIO" instruction (MONIO is a symbolic label for a four position area in which the monitor builds an I/O and branch instruction). "MONTOR" (MONTOR is a symbolic label in the monitor program for a branch to the library data transmission routine) then checks the data terminal in the library. If the data terminal is busy reading, polling (waiting) or transmitting, it is ignored and the program branches to test the switches. If the data terminal is not busy, but dead, a poll list is sent (asking if the library wants on) and then branches back to MONTOR. If the data terminal is not dead and a service request is being transmitted from the library, the library data are processed immediately and the appropriate message is transmitted back to the library by underground telephone cable. MONTOR is executed again to assure the message is being transmitted. If the data terminal is busy the program will ignore it and move to test the switches. If the data terminal is busy, although not dead, and there is no service request and no bid for the line, then there is trouble, and the program branches to MONTOR for a recheck.

The second part of the program works the same as the stop button on other computers. Since the 1401 must always be on to process the library, the F and G switches are used as a waiting place. The G switch is tested first. If it

is reverse from its last status, the next program from the user is loaded. If the G switch has not been reversed but the F switch has, the program waits until the F switch is reversed manually and then continues. This switch is mainly for the operator's convenience; it allows him to stop whenever necessary.

The third part of the program is to determine what the I/O is. If the I/O type is disk, then the instruction is to read and write disk. If the I/O is equal to F (punch-read-feed) then the punch-read-feed is set up. If the I/O is not one of the above, the I/O originally stored is executed and control is returned to the user program.

If the I/O is the user program is a no op, the program branches back to MONTOR to wait. The internal F switch is reversed to wait if the I/O is not a no op. (The operator has pushed start-reset-start physically.) Until the F switch is reversed externally, the operation is waiting and then it goes back through the program.